

# **DESENVOLVIMENTO DE UMA PLATAFORMA PARA SIMULAÇÃO DE SATÉLITES**

Angelo dos Santos Lunardi<sup>1</sup>; Rodrigo Alvite Romano<sup>2</sup>.

<sup>1</sup> Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

<sup>2</sup> Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

**Resumo.** *Com a disponibilidade crescente de recursos computacionais e da evolução das ferramentas de simulação, cada vez mais a concepção de sistemas é realizada com o auxílio de simuladores. No caso da área aeroespacial, devido ao alto custo e à complexidade dos problemas envolvidos, plataformas de simulação são elementos indispensáveis em qualquer projeto. Este trabalho de iniciação científica visa o desenvolvimento de um simulador para a dinâmica de atitude (orientação) de um satélite. Um simulador consiste no cálculo da orientação e na representação gráfica do mesmo. Para isso conforme a orientação é calculada a partir dos torques gerados por atuadores, um visualizador representa essa dinâmica através de um desenho tridimensional, tudo em tempo-real, isso para que seja mais clara como a dinâmica do satélite se comporta ao longo do tempo.*

## **Introdução**

No contexto da área aeroespacial, simuladores são ferramentas de software que podem ser utilizados por pesquisadores, engenheiros, alunos ou mesmo gerentes para analisar e avaliar as operações de aeronaves, satélites e veículos lançadores. As informações obtidas fornecem recursos que auxiliam a responder perguntas a respeito de um projeto ou produto. Esses simuladores podem ser desenvolvidos a partir de requisitos bastante específicos de uma determinada missão, ou então, para uma classe mais generalizada de aeronaves. Em geral, o código do simulador é composto por ferramentas associadas (que realizam operações de codificação, tais como bibliotecas de álgebra linear, integradores numéricos, ou pacotes para cálculo de órbita) e por algoritmos derivados. Normalmente, o software do simulador é executado com determinadas condições iniciais e operacionais durante o período de tempo necessário para analisar um conjunto de características. Por fim, os dados são exibidos ao usuário, no formato de texto, ou através de interfaces gráficas.

Uma evidência da importância das ferramentas de software na área aeroespacial é a quantidade de simuladores disponíveis. Há pacotes de simulação para uma ampla gama de funções que são úteis para os engenheiros e cientistas. Dentre tais funcionalidades, pode-se citar: análise de órbita e de atitude, testes hardware-in-the-loop (HIL) e verificação/validation do sistema de controle. Além dos pacotes gratuitos, há opções comerciais submetidas a procedimentos rigorosos de validação que geralmente são bem aceitos pela comunidade de engenheiros e cientistas. O grau de funcionalidade varia muito entre os pacotes disponíveis, podendo-se encontrar desde simples opções para reproduzir a propagação de dois corpos representados por pontos de massa até complexos simuladores tridimensionais de satélites de comunicação.

Este trabalho visa desenvolver um simulador para a dinâmica de orientação (atitude) de um satélite em relação a um eixo referência. Como os dados de dinâmica de atitude são tridimensionais, pretende-se disponibilizar o recurso de visualização em três dimensões, de modo a oferecer uma plataforma mais lúdica para a interpretação de resultados das simulações.

A primeira etapa da pesquisa foi realizar a busca por um simulador de satélites. Desta etapa, optou-se por empregar um modelo da plataforma multimissão (PMM) (Santana, 2011) que vem sendo desenvolvida através do programa espacial brasileiro. A segunda etapa atende ao módulo de visualização tridimensional da orientação do satélite. Para tanto, utilizou-se a API gráfica OpenGL (Scalco, 2005) que oferece recursos para a elaboração de inúmeros desenhos tridimensionais. Assim, tem-se um computador (PC1) que reproduz a dinâmica de atitude, e outro (PC2) que recebe a atitude através dos ângulos de Euler ( $\varphi, \theta, \psi$ ) e gera os gráficos correspondentes.

## Material e Métodos

### Modelo matemático da dinâmica de atitude.

Nesse trabalho utilizou-se a representação por ângulos de Euler para descrever a dinâmica de atitude do satélite. A orientação de um corpo rígido em relação a um referencial fixo pode ser representada por meio uma sequência de três rotações (nesse caso adotou-se uma sequência onde se aplica primeiro a rotação em torno do eixo z em seguida y e x, onde usualmente é denotado como sequência é 3-2-1). As matrizes de rotação em relação aos eixos z, y, x são dadas por

$$\begin{aligned} R_z \psi &= \begin{matrix} c \psi & s \psi & 0 \\ -s \psi & c \psi & 0 \\ 0 & 0 & 1 \end{matrix} \\ R_y \theta &= \begin{matrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{matrix} \\ R_x \varphi &= \begin{matrix} 1 & 0 & 0 \\ 0 & c \varphi & s \varphi \\ 0 & -s \varphi & c \varphi \end{matrix}, \end{aligned}$$

respectivamente (Crassidis, 2011). Os operadores c . e s . denotam as operações de coseno e seno. Logo, a orientação do corpo em relação ao referencial fixo é obtida através da matriz de rotação

$$R = R_x \varphi R_y \theta R_z \psi = \begin{matrix} c \psi & c(\theta) & s \psi & c(\theta) & -s(\theta) \\ -c \varphi & s \psi + s \varphi s(\theta) c \psi & c \varphi s \psi + s \varphi s(\theta) s \psi & s \varphi c \theta & s \varphi c \theta \\ s \varphi s \psi + c \varphi s(\theta) c \psi & -s \varphi c \psi + c \varphi s(\theta) s \psi & c \varphi c \theta & c \varphi c \theta & c \varphi c \theta \end{matrix}$$

O modelo da dinâmica de atitude da plataforma multimissão em torno do ponto de equilíbrio em (Santana, 2008). Tal dinâmica pode ser representada pela equação de estados.

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t),$$

em que  $x(t) = \varphi \quad \theta \quad \psi \quad \dot{\varphi} \quad \dot{\theta} \quad \dot{\psi}^T$  denota o vetor de estados composto pelos ângulos de Euler e pelas suas respectivas derivadas no tempo. O vetor de entrada é formado pelos torques em relação a cada eixo, ou seja,  $u(t) = u_1 \quad u_2 \quad u_3^T$ .

As matrizes A e B na equação de estados assumem os valores:

$$A = \begin{matrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{4\omega_o^2(J_z - J_y)}{J_x} & 0 & 0 & 0 & 0 & \frac{\omega_o(J_x - J_y + J_z)}{J_x} \\ 0 & \frac{3\omega_o^2(J_x - J_z)}{J_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\omega_o^2(J_x - J_y)}{J_z} & \frac{\omega_o(J_y - J_x - J_z)}{J_z} & 0 & 0 \end{matrix}$$
  

$$B = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{l}{J_x} & 0 & 0 \\ 0 & \frac{l}{J_y} & 0 \\ 0 & 0 & \frac{l}{J_z} \end{matrix}$$

Assim, a equação de saída do modelo é dada por:

$$y(t) = C \cdot x(t)$$

Com

$$C = \begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$$

Parâmetros do modelo são apresentados na Tabela 1.

Tabela 1 – Parâmetros do Modelo da PMM

Parâmetros	Valores
Momento de inercia (sem carga) (Kgm <sup>2</sup> )	J <sub>x</sub> =305,891 J <sub>y</sub> =314,064 J <sub>z</sub> =167,339
Braço de Torque(m)	<i>l</i> =1
Velocidade angular em orbita (rad.s <sup>-1</sup> )	ω <sub>o</sub> =0.001
Massa (Kg)	578,052
Altura da orbita (Km)	750

### Sistema de Controle.

Como a dinâmica de atitude é inherentemente instável em malhar aberta, projetou-se um controlador por realimentação de estados (Ogata, 2010) e um observador, para manter o satélite estável na orientação pré-determinada:  $y(t) = [0^\circ \ 0^\circ \ 0^\circ]^T$ . Portanto, os torques aplicados ao satélite são determinados através da lei de controle

$$u(t) = -Kx(t),$$

a matriz de ganho  $K$  é calculada de modo a resolver o funcional

$$J = \int x^T(t) Q x(t) - u^T(t) R u(t) dt,$$

cuja solução é dada pela equação de Riccati

$$A^T P + PA - PBR^{-1}B^T P + Q = 0.$$

Note que é necessário se definir as matrizes  $Q$  e  $R$ , que ponderam o erro no estado e o esforço de controle, respectivamente. Nesse caso, obteve-se após algumas simulações os seguintes valores

$$Q = \begin{matrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{matrix} \quad R = \begin{matrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Tais valores foram adotados, pois respeitam a saturação de 0.1 Nm de esforço na entrada e o erro de 0.1° em cada eixo como tolerância de apontamento fino.

Depois de determinar a matriz  $K$ , projetou-se um observador de Luenberger (Ogata, 2010) para estimar o vetor de estados  $x(t)$ . Portanto, a dinâmica desse estimador é dado por

$$\begin{aligned} \dot{x}(t) &= A' \cdot x(t) + B' \cdot u(t) + L \cdot y(t) - y(t) \\ y(t) &= C' \cdot x(t) \end{aligned}$$

A matriz  $L$  é usada para corrigir o erro do estimador e foi obtida por alocação de polos, de modo que os polos do observador sejam seis vezes mais rápidos que os polos da planta. A ligação entre as matrizes da PMM e do controlador mais observador podem ser demonstrada conforme o diagrama abaixo (Figura 1).

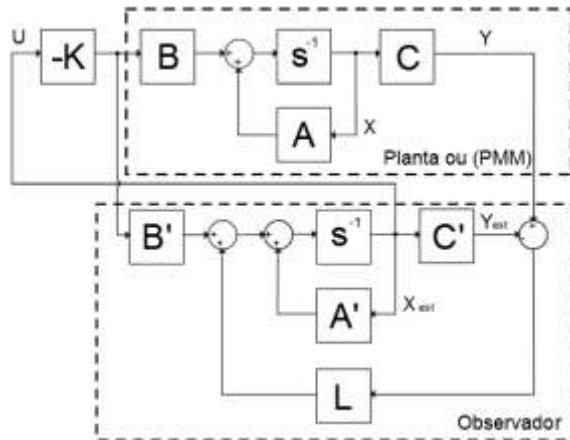


Figura 1 - Diagrama Planta mais Controlador.

Como se pode observar, as equações do observador utiliza um conjunto de matrizes diferentes das usadas no modelo de simulação. Com isso, é possível testar aspectos de robustez em situações de erros de modelagem representados por incertezas estruturadas. Por essa razão, utiliza-se uma nomenclatura diferenciada para as matrizes do modelo.

### Subsistema de visualização tridimensional

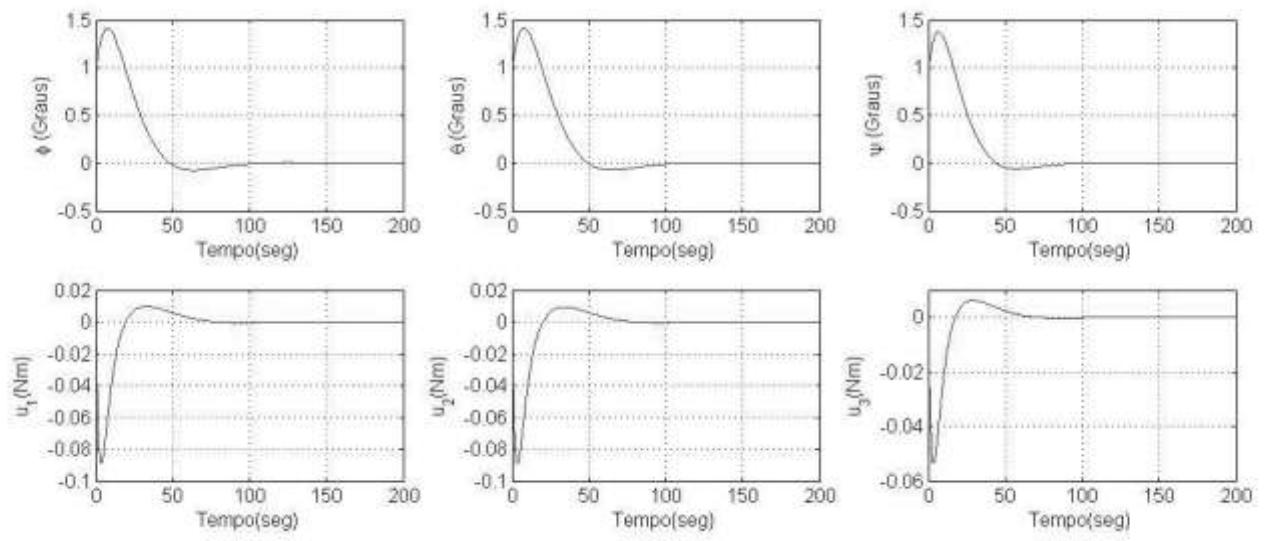
Os gráficos gerados são representados por um desenho tridimensional desenvolvido no OpenGL, a criação desse objeto tridimensional consiste em uma programação em linhas de código C++, em que determinando vértices, inclinações, cores, luzes e entre outros elementos, assim é gerado o desenho.

Para isso utilizou-se o Embarcadero® C++Builder® XE, um compilador C++ que permite utilizar as API gráficas do OpenGL, essa API possui funções que permitem desenvolver qualquer desenho 2D ou 3D. No projeto como o objeto em estudo é um satélite e sua orientação tridimensional, utilizando uma combinação de superfícies planas para gerar o corpo do satélite e o painel solar, e algumas funções próprias da biblioteca do OpenGL chamada GLUT (*OpenGL Utility Toolkit*), em que os desenhos de uma esfera ou de um cilindro já estão pronto. Porém para gerar uma semiesfera foi preciso desenvolver uma função específica utilizando as funções básicas do OpenGL, por exemplo, foi preciso determinar um conjunto de vértices em função do raio e em loop fecham uma superfície semelhante a uma semiesfera.

Como já se utilizou o Embarcadero® C++Builder® XE para se desenvolver o desenho, também se utilizou a API do Windows para comunicação serial. No Embarcadero® C++Builder® XE pode-se configurar a porta desejada e o baud rate no caso 115200 bps, para suportar a taxa de amostragem dos dados enviados pelo MATLAB®Simulink, que também por sua vez envia pela serial com a mesma taxa. Para controlar essa taxa o MATLAB®Simulink oferece uma função chamada Real-Time Windows Target, com ela pode-se ter a amostra enviada a cada 0,1s, e o OpenGL atualizando automaticamente o desenho.

## **Resultados e Discussão**

O funcionamento do sistema é avaliado através de um ensaio em que, inicialmente, a PMM está apontada para a direção  $\varphi \quad \theta \quad \psi^T = 1^\circ \quad 1^\circ \quad 1^\circ T$ . Com tendência de afastamento de  $\omega=0,1 \text{ rad.s}^{-1}$  em relação à orientação desejada:  $0^\circ \quad 0^\circ \quad 0^\circ T$ . Os resultados obtidos estão apresentados na Figura 2, que mostram um pico um pouco maior que 0.08 Nm nos atuadores que corrigem os ângulos  $\varphi$  e  $\theta$ , já no ângulo  $\psi$  vemos que o bico que menos devido a um valor menor de momento de inercia, nota-se também um erro estacionário nulo.



Para um maior entendimento do sistema realizou-se mais dois ensaios, agora com condição inicial  $\varphi \ \theta \ \psi^T = 10^\circ \ 10^\circ \ 10^\circ T$ , porém mantendo a tendência de afastamento e respeitando a saturação da roda em 0.1 Nm (Carrara, 2010). Os resultados obtidos estão apresentados na Figura 3 que mostram uma saturação nos atuadores de 0.1 Nm por um período de tempo em torno de 25 segundos para que o satélite seja desacelerado e comece a corrigir sua posição, como no primeiro ensaio o erro estacionário nulo se mantém.

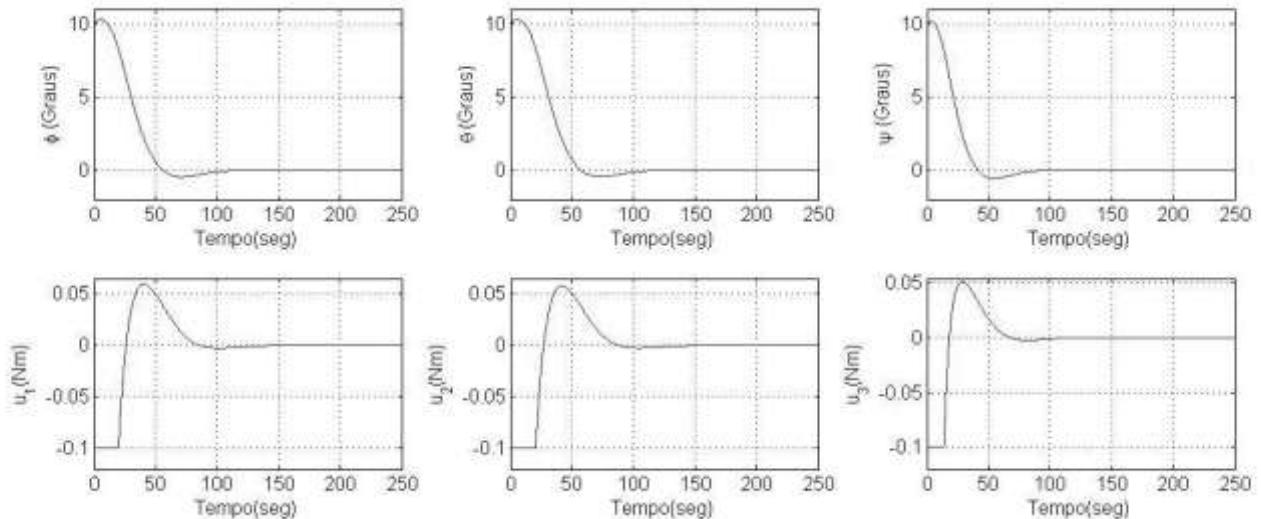


Figura 3 - Ensaio 2

Já o terceiro ensaio com condições iniciais de  $\varphi \ \theta \ \psi^T = 5^\circ \ 5^\circ \ 5^\circ T$ , porém, agora com um ruindo branco Gaussiano, com média nula e uma variância em cada eixo de  $10^{-4}$ . Os resultados obtidos estão representados na Figura 4 que mostram devido ao ruído uma grande oscilação nos torques aplicados em cada eixo porém não excedendo o torque máximo de 0.1 Nm, já nesse caso há um erro estacionário, porém esse erro ainda respeita a tolerância de um apontamento fino de  $0.1^\circ$ .

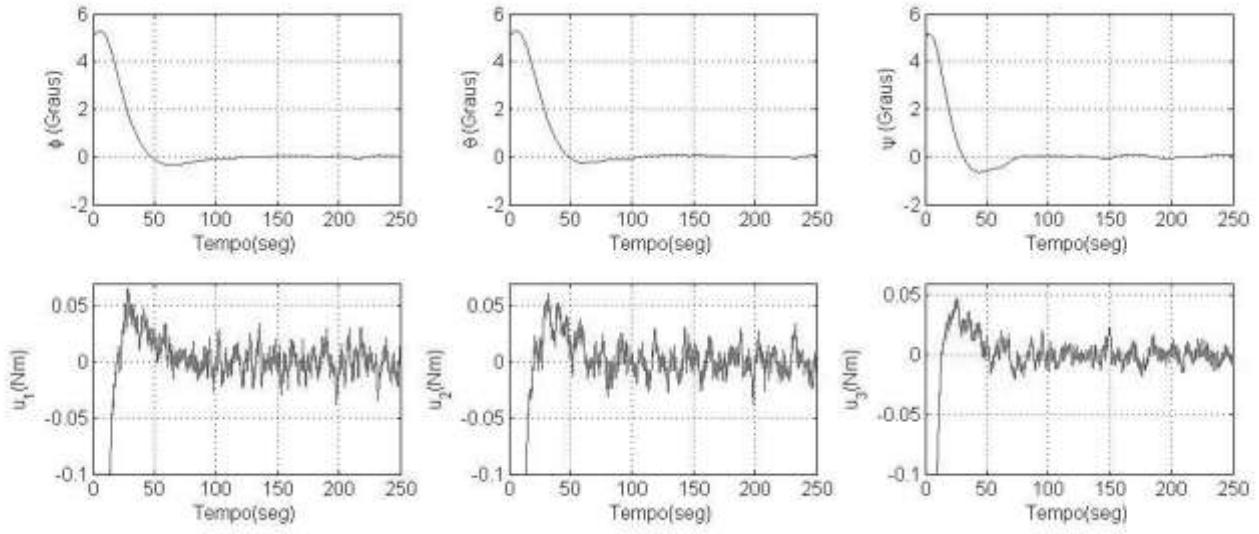


Figura 4 - Ensaio 3

Nota-se que com pequenos erros de posição e respeitando as especificações da roda de reação, em ambos os ensaios levam em media 100s para estabilizar no  $0^\circ \ 0^\circ \ 0^\circ {}^T$ , porém com o ruído esse erro nulo não é alcançado, mas há uma tolerância de  $\varphi \ \theta \ \psi {}^T = 0.1^\circ \ 0.1^\circ \ 0.1^\circ {}^T$ . (Amaral, 2008) para a posição no apontamento fino. No projeto certos aspectos foram desconsiderados (ruído de sensoriamento, efeitos radioativos, entre outros), devido a complexidade de se caracterizar tais fenômenos, com tudo, é visível a atuação do controle regulatório sobre a planta mesmo com o observador estimando os estados.

Após os ensaios, conforme a orientação é modificada, o comportamento correspondente é ilustrado através de um desenho tridimensional, a Figura 6 demonstra a posição inicial, no caso fora da posição pré-determinada semelhante ao ensaio dois, e a Figura 7 a posição final com o satélite estabilizado.

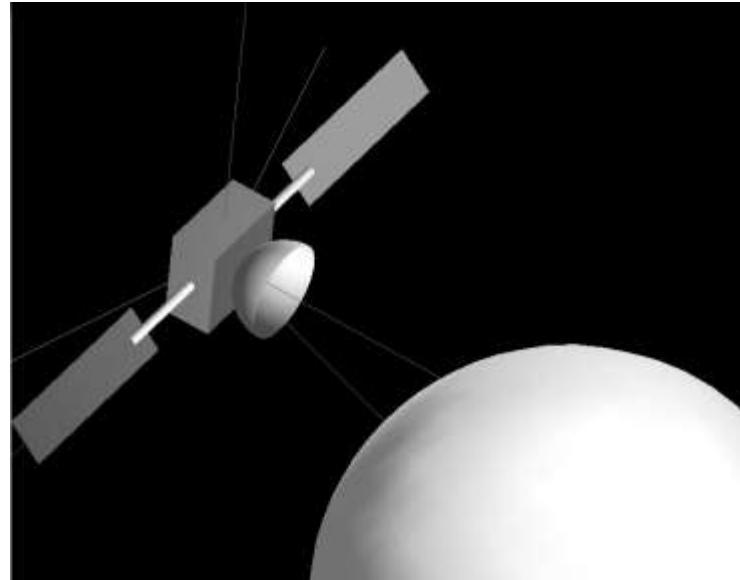


Figura 5 - Visualizador 3D

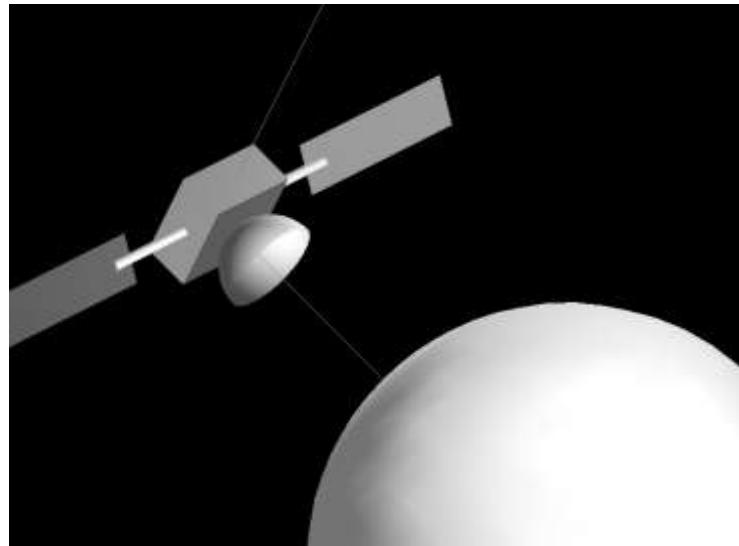


Figura 6 - Visualizador 3D

## Conclusões

A principal contribuição dessa pesquisa foi o desenvolvimento da plataforma tridimensional utilizando recurso computacionais como OpenGL para o desenho tridimensional e o MATLAB®Simulink para realizar os cálculos necessários da dinâmica do satélite. Com o visualizador tridimensional o entendimento da dinâmica de orientação é mais clara, em comparação aos dados representados por gráficos. Isso se da devido à visualização da atitude, a composição dos três eixos representados por um único desenho. Com isso algumas características podem ser melhor visualizadas como o próprio apontamento, a velocidade ou como a dinâmica se comporta ao longo do tempo. Certamente isso é devido a todo o sistema de simulação e visualização estar em tempo real, caso contrário não teria um sentido físico se a dinâmica acontecesse no tempo em que o computador a calculasse, pois há elementos mecânicos envolvidos que não respondem instantaneamente.

O OpenGL associado a softwares de simulação de sistemas dinâmicos oferece uma plataforma que permite o desenvolvimento de simuladores com recursos de visualização tridimensionais ou bidimensionais. Esta plataforma pode ser explorada na simulação de inúmeros dispositivos, por exemplo: VANTs (Veículos aéreos não tripulados) ou sistemas eletromecânicos. Deste modo, facilita-se o desenvolvimento de controladores para esses dispositivos, uma vez que é possível avaliar o comportamento desses sistemas de uma forma muito mais intuitiva. Com situações em que o experimento físico seria complicado ou até impossível, os simuladores auxiliam na compreensão do sistema real, possibilitando até antecipar possíveis erros, como mau dimensionamento, material não adequando, entre outros.

O OpenGL e o MATLAB®Simulink mesmo sendo dois software distintos através da comunicação serial podem trabalhar juntos se tornando uma ferramenta muito útil para realizar controle quando não se tem a planta física. Essa associação só é possível por que todo o sistema esta em tempo real, a única limitação é devido à taxa de amostragem tem que ser maior do que o tempo total que o MATLAB®Simulink calcula a dinâmica do satélite.

Uma possível continuação para esse projeto é ao invés de usar um único elemento para simular a orientação da PMM, pretende-se ainda incorporar ao sistema um módulo de controle embarcado. Nessa configuração os dados do simulador seriam enviados ao módulo embarcado que, a partir dessa informação, gera o torque necessário para manter o satélite em uma orientação pré-determinada.

## Referências Bibliográficas

- Amaral, J. C., (2008), *Análise, projeto e simulação de uma arquitetura de controle configurável para a plataforma multimissão*.
- Carrara, V., (2010), *Comparação experimental entre formas de controle de atitude com rodas de reação*. VI SBEIN - Simpósio Brasileiro de Engenharia Inercial. Rio de Janeiro.
- Crassidis, J. L., & Junkins, J. L. (2011). *Optimal estimation of dynamic systems* (Vol. 24). CRC press.
- Ogata,K. (2010) *Engenharia de controle moderno*. 5<sup>a</sup> edição. São Paulo, Prentice Hall.
- Santana, A. C., (2011) Attitude Stabilization of the PMM Satellite Using a LQG-Based Control Strategy doi: 10.5540/Tema. 2008.09.02.0321. *TEMA-Tendências em Matemática Aplicada e Computacional*, **9.2**, 321-330.
- Scalco, R. (2005) *Introdução à Biblioteca Gráfica*. (Nota de Aula).